

```

1 <-- DML, DATA MANAGE LANGUAGE (INSERT, UPDATE, DELETE)
2 -- UPDATE : 테이블에 포함된 기존 데이터를 수정 (데이터 건 수는 달라지지 않음)
3 /*
4 UPDATE TABLE_NAME
5 SET [COLUMN = VALUE,] cf. VALUE에 서브쿼리가 들어갈 수 있다.
6 WHERE 기본키 컬럼 = VALUE;
7
8 */
9
10
11 SELECT *
12 FROM DEPARTMENT;
13
14 UPDATE DEPARTMENT
15 SET DEPT_NAME = '전략기획팀'
16 WHERE DEPT_ID = '90';
17
18
19 /* 심하균의 부서배치, 직급 수정 -> 성해교의 부서배치, 직급과 동일하게 변경 */
20
21 SELECT *
22 FROM EMPLOYEE
23 WHERE EMP_NAME = '감우섭';
24
25
26 SELECT JOB_ID, DEPT_ID
27 FROM EMPLOYEE
28 WHERE EMP_NAME = '성해교';
29
30 UPDATE EMPLOYEE
31 SET (JOB_ID, DEPT_ID) = (SELECT JOB_ID, DEPT_ID
32 FROM EMPLOYEE
33 WHERE EMP_NAME = '성해교')
34 WHERE EMP_NAME = '심하균';
35
36
37 UPDATE EMPLOYEE
38 SET MARRIAGE = DEFAULT
39 WHERE EMP_NAME = '감우섭'; /* 테이블 생성시 DEFAULT 값이 있는 경우
40
41 -- UPDATE와 무결성 제약조건
42 -- 변경하고자 하는 값이 제약에 걸려있는 컬럼인지 확인 필요 Q
43
44 UPDATE EMPLOYEE
45 SET DEPT_ID = '55'
46 WHERE EMP_NAME = '감우섭';
47
48 -- DELETE : 테이블에 포함된 기존 데이터를 삭제

```

```

49 -- 행 단위로 삭제되므로 전체 행 수가 달라진다 (테이블의 구조는 유지됨)
50 -- DELETE와 무결성 제약조건
51
52
53 /*
54 DELETE [FROM] TABLE_NAME
55 WHERE CONDITION ; * WHERE가 없을 경우 전체 행 삭제됨. 서브쿼리 사용 가능
56
57 TRUNCATE TABLE TABLE_NAME ; (전체 행을 삭제할 때. DML은 원복이 가능하지만,
58
59 */
60
61
62 /*삭제 불가*/
63 DELETE FROM DEPARTMENT
64 WHERE DEPT_ID = '90';
65
66 SELECT *
67 FROM EMPLOYEE;
68
69 DELETE FROM EMPLOYEE
70 WHERE EMP_NAME = '이중기';
71
72
73 /*
74 -
75 -DDL
76 -DML
77 -TCL
78 -
79 */
80
81 -- TCL (TRANSACTION CONTROLLER LANGUAGE) : COMMIT, ROLLBACK
82 -- DML 구문 (INSERT, UPDATE, DELETE) 이 시작될 때 시작
83 -- cf. COMMIT 친 이후에는 ROLLBACK 불가
84 -- 데이터 일관성을 유지하기 위한 목적으로 사용하는 논리적으로 연관된 작업의 집합
85
86
87
88 -- VIEW (STORED QUERY, VIRTUAL TABLE)
89 -- 하나 또는 하나 이상의 테이블에 포함된 데이터 부분 집합을 나타내는 논리적 테이블
90 -- DML 불가. 읽기 전용의 데이터 생성
91
92
93 /*
94 <뷰 생성>
95 CREATE OR REPLACE VIEW VIEW_NAME
96 AS
97 SUBQUERY;
98

```

99<뷰 삭제>

2025년 08월 08일 10:01 오후 4:00

```
100 DROP VIEW VIEW_NAME;
101 */
102
103
104 CREATE OR REPLACE VIEW VIEW_NAME
105 AS
106 SELECT EMP_NAME, DEPT_ID, PHONE
107 FROM EMPLOYEE
108 WHERE DEPT_ID = '90';
109
110 SELECT *
111 FROM VIEW_NAME;
112
113 /* 재사용 가능한 논리적 테이블 생성 */
114 CREATE OR REPLACE VIEW V_EMP_DEPT
115 AS
116 SELECT EMP_NAME AS NAME, DEPT_NAME AS DEPT
117 FROM EMPLOYEE E
118 JOIN DEPARTMENT D USING (DEPT_ID);
119
120 SELECT *
121 FROM V_EMP_DEPT;
122
123 -- 인 라인뷰 - TOP N 분석 개념
124 -- 인 라인뷰 필요
125 -- 조건에 맞는 최상위 (최하위) 레코드 N개를 식별해야하는 경우
126
127 /*
128 1. 정렬
129 2. ROWNUM 정렬된 데이터 레코드에 순번 부여
130 3. 부여된 순번을 이용해서 필요한 수 만큼 식별
131 */
132
133 SELECT ROWID, ROWNUM, EMP_NAME /* 가상 컬럼. DBMS가 식별할 수 있는 키.
134 FROM EMPLOYEE;
135
136 -- 급여 평균보다 많은 급여를 받는 사원 정보 조회
137 -- 가장 급여가 높은 사원 1명 식별
138 -- 또는 상위 3명 식별
139
140 SELECT ROWNUM, EMP_NAME, SALARY
141 FROM (SELECT EMP_NAME, SALARY
142 FROM (SELECT DEPT_ID AS "DID",
143 ROUND(AVG(SALARY), -3) AS "DAVG"
144 FROM EMPLOYEE
145 GROUP BY DEPT_ID) V
146 JOIN EMPLOYEE ON (NVL(DEPT_ID, 'N/A') = V.DID)
```

```

147 WHERE SALARY > V.DAVG
148 ORDER BY SALARY DESC)
149 WHERE ROWNUM <= 5;
150
151
152 -- WINDOW FUNCTION (순위, 집계)
153 -- RANK() ~ WITHIN GROUP
154 -- RANK() OVER(그룹화할 대상)
155
156 SELECT RANK(2300000) WITHIN GROUP (ORDER BY SALARY DESC) AS R
157 FROM EMPLOYEE;
158
159
160
161 SELECT *
162 FROM ( SELECT EMP_NAME, SALARY,
163          RANK() OVER (ORDER BY SALARY DESC) AS R
164        FROM EMPLOYEE )
165 WHERE R = 3;
166
167 -----DDL 실습
168
169
170 /* 1. 다음 설정을 참조하여 테이블을 생성하시오. */
171
172 -- CUSTOMERS 테이블
173 DROP TABLE CUSTOMERS;
174
175 CREATE TABLE CUSTOMERS (
176 CNO NUMBER(5) PRIMARY KEY,
177 CNAME VARCHAR2(10) NOT NULL,
178 ADDRESS VARCHAR2(50) NOT NULL,
179 EMAIL VARCHAR2(20) NOT NULL,
180 PHONE VARCHAR2(20) NOT NULL
181 );
182
183 SELECT *
184 FROM CUSTOMERS;
185
186 INSERT INTO CUSTOMERS
187 INSERT INTO CUSTOMERS
188
189
190 -- PRODUCTS 테이블
191 DROP TABLE PRODUCTS;
192
193 CREATE TABLE PRODUCTS (
194 PNO NUMBER(5) PRIMARY KEY,

```

```

195 PNAME VARCHAR2(20) NOT NULL,
196 COST NUMBER(8) NOT NULL,
197 STOCK NUMBER(5) NOT NULL
198 );
199
200 SELECT *
201 FROM PRODUCTS;
202
203
204
205 -- ORDERS 테이블
206 DROP TABLE ORDERS CASCADE CONSTRAINTS;
207
208 CREATE TABLE ORDERS (
209 ORDERNO NUMBER(10) PRIMARY KEY,
210 ORDERDATE date DEFAULT SYSDATE NOT NULL ,
211 ADDRESS VARCHAR2(50) NOT NULL,
212 PHONE VARCHAR2(20) NOT NULL,
213 STATUS VARCHAR2(20) NOT NULL CHECK (STATUS IN ('결제완료', '배송중',
214 CNO NUMBER(5) NOT NULL,
215 FOREIGN KEY (CNO) REFERENCES CUSTOMERS (CNO)
216 );
217
218 SELECT *
219 FROM ORDERS;
220
221 INSERT INTO ORDERS (ORDERNO, CNO, STATUS)
222 VALUES ('002', '551', '결제완료');
223
224
225 -- ORDERDETAIL 테이블
226
227 DROP TABLE ORDERDETAIL;
228
229 CREATE TABLE ORDERDETAIL (
230 ORDERNO NUMBER(5),
231 PNO NUMBER(20),
232 QTY NUMBER(8) DEFAULT 0,
233 COST NUMBER(5) DEFAULT 0,
234 PRIMARY KEY (ORDERNO, PNO),
235 FOREIGN KEY (ORDERNO) REFERENCES ORDERS (ORDERNO),
236 FOREIGN KEY (PNO) REFERENCES PRODUCTS (PNO)
237 );
238
239
240 SELECT *
241 FROM ORDERDETAIL;
242

```

```

243< -- 05 03 00 -- -- 0005년 08월 0001 7001 0월 4 00
244 /*2. products 테이블에 다음 데이터를 입력하시오.*/
245
246 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
247 VALUES (1001, '삼양라면', '1000', '1000') ;
248
249
250 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
251 VALUES (1002, '새우깡', '1500', '500') ;
252
253
254 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
255 VALUES (1003, '월드콘', '2000', '350') ;
256
257
258 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
259 VALUES (1004, '빼빼로', '2000', '700') ;
260
261 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
262 VALUES (100, '코카콜라', '1800', '550') ;
263
264 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
265 VALUES (1006, '환타', '1600', '300') ;
266
267 SELECT *
268 FROM PRODUCTS ;
269
270
271 /*3. customers 테이블에 다음 데이터를 입력하시오.*/
272 INSERT INTO CUSTOMERS (CNO, CNAME, ADDRESS, EMAIL, PHONE)
273 VALUES (101, '김철수', '서울 강남구', 'cskim@naver.com', '899-6666') ;
274
275 INSERT INTO CUSTOMERS (CNO, CNAME, ADDRESS, EMAIL, PHONE)
276 VALUES (102, '이영희', '부산 서면', 'yhlee@empal.com', '355-8882') ;
277
278 INSERT INTO CUSTOMERS (CNO, CNAME, ADDRESS, EMAIL, PHONE)
279 VALUES (103, '최진국', '제주 동광양', 'jkchoi@gmail.com', '852-5764') ;
280
281 INSERT INTO CUSTOMERS (CNO, CNAME, ADDRESS, EMAIL, PHONE)
282 VALUES (104, '강준호', '강릉 홍제동', 'jhkang@hanmail.com ', '559-
283
284 INSERT INTO CUSTOMERS (CNO, CNAME, ADDRESS, EMAIL, PHONE)
285 VALUES (105, '민병국', '대전 전민동', 'bgmin@hotmail.com ', '559-
286
287 INSERT INTO CUSTOMERS (CNO, CNAME, ADDRESS, EMAIL, PHONE)
288 VALUES (106, '오민수', '광주 북구', 'msoh@microsoft.com', '542-9988') ;
289
290 SELECT *

```

```

291<---\ 25 02 00          2005년 2월 28일 금요일 오후 4:20
FROMCUSTOMERS;
292
293 /* 4. 다음과 같은 주문 정보를 orders 테이블과 orderdetail 테이블에 입력하시오.
294 검색하여 입력할 것. orders에 1건, orderdetail에 1건을 입력한다.
295 "김철수(101)가 3일전에 삼양라면(1001)을 개당 1000원에 50개 주문하였다." */
296
297 SELECT*
298 FROMORDERS;
299
300 SELECT*
301 FROMORDERDETAIL;
302
303 SELECT*
304 FROMCUSTOMERS;
305
306 SELECT*
307 FROMPRODUCTS;
308
309 INSERT INTO ORDERS (ORDERNO, ORDERDATE, ADDRESS, PHONE,
310 VALUES (1, SYSDATE-3, '서울 강남구', '899-6666', '결제완료', '101');
311
312 INSERT INTO ORDERDETAIL (ORDERNO, PNO, QTY, COST)
313 VALUES (1, '1001', '50', '1000');
314
315
316
317 /* 5. 위와 같은 주문 정보에서 해당 상품(products)의 재고(stock)을 수정하시오.
318 "삼양라면(1001)의 재고를 150(=200-50)개로 변경한다" */
319
320 /*
321 UPDATE TABLE_NAME
322 SET [COLUMN = VALUE,] cf. VALUE에 서브쿼리가 들어갈 수 있다.
323 WHERE기본키 컬럼 = VALUE;
324
325 */
326
327 SELECT*
328 FROMPRODUCTS;
329
330 UPDATEPRODUCTS
331 SET STOCK = '150'
332 WHERE PNO = '1001';
333
334
335 /*6. 다음과 같은 주문 정보를 orders 테이블과 orderdetail 테이블에 입력하시오.
336 검색하여 입력할 것. orders에 1건, orderdetail에 2건을 입력한다.
337 "이영희(102)가 이틀전에 새우깡(1002)을 개당 1500원에 100개, 월드콘(1003)을
338 주문하였다." */
339

```

```

340<---\ 25 02 00
340 SELECT *
341 FROM ORDERS;
342
343 SELECT *
344 FROM CUSTOMERS;
345
346 SELECT *
347 FROM ORDERDETAIL;
348
349 SELECT *
350 FROM PRODUCTS;
351
352
353
354 INSERT INTO ORDERS (ORDERNO, ORDERDATE, ADDRESS, PHONE,
355 VALUES (2, SYSDATE-2, '부산 수영구', '337-5000', '결제완료', '102' );
356
357 INSERT INTO ORDERDETAIL (ORDERNO, PNO, QTY, COST)
358 VALUES (2, '1002', 100, 1500);
359
360
361 INSERT INTO ORDERDETAIL (ORDERNO, PNO, QTY, COST)
362 VALUES (2, '1003', 150, 2000);
363
364
365 /*7. 위와 같은 주문 정보에서 해당 상품(products)의 재고(stock)을 수정하시오.
366  "새우깡(1002)의 재고를 400(=500-100)개로 변경한다"
367  "월드콘(1003)의 재고를 200(=350-150)개로 변경한다" */
368
369 SELECT *
370 FROM PRODUCTS;
371
372 UPDATE PRODUCTS
373 SET STOCK = '400'
374 WHERE PNO = '1002';
375
376 UPDATE PRODUCTS
377 SET STOCK = '200'
378 WHERE PNO = '1003';
379
380 /*8. 다음과 같은 주문 정보를 orders 테이블과 orderdetail 테이블에 입력하시오.
381 검색하여 입력할 것. orders에 1건, orderdetail에 2건을 입력한다.
382 "오민수(106)가 어제 빼빼로(1004)를 개당 2000원에 100개, 코카콜라(1005)를 개당
383
384 SELECT *
385 FROM ORDERS;
386
387 SELECT *

```



```

388 FROM ORDERDETAIL;
389
390 SELECT *
391 FROM CUSTOMERS;
392
393 SELECT *
394 FROM PRODUCTS;
395
396 INSERT INTO ORDERS (ORDERNO, ORDERDATE, ADDRESS, PHONE,
397 VALUES (3, SYSDATE-1, '광주 북구', '652-2277', '결제완료', '106' );
398
399 INSERT INTO ORDERDETAIL (ORDERNO, PNO, QTY, COST)
400 VALUES (3, '1004', '100', '2000');
401
402 INSERT INTO ORDERDETAIL (ORDERNO, PNO, QTY, COST)
403 VALUES (3, '1005', '50', '1800');
404
405 /*9. 위와 같은 주문 정보에서 해당 상품(products)의 재고(stock)을 수정하시오.
406  ``빠빠로(1004)의 재고를 600(=700-100)개로 변경한다``
407  ``코카콜라(1005)의 재고를 500(=550-50)개로 변경한다``*/
408
409 SELECT *
410 FROM PRODUCTS;
411
412 UPDATE PRODUCTS
413 SET STOCK = '600'
414 WHERE PNO = '1004';
415
416
417 UPDATE PRODUCTS
418 SET STOCK = '500'
419 WHERE PNO = '1005';
420
421
422 /*10. 다음과 같은 전체 주문 목록을 출력하는 문장을 작성하시오.*/
423
424 SELECT O.ORDERDATE, C.CNAME, C.ADDRESS, C.PHONE, O.STATUS,
425 FROM CUSTOMERS C
426 JOIN ORDERS O ON (C.CNO = O.CNO)
427 JOIN ORDERDETAIL D USING (ORDERNO)
428 JOIN PRODUCTS P USING (PNO);
429
430
431 /*11. 다음과 같이 일별 매출 목록을 출력하는 문장을 작성하시오.*/
432
433
434
435 SELECT ORDERDATE, SUM(COST*QTY)

```

```

436 FROM ORDERS O
437 JOIN ORDERDETAIL D ON (D.ORDERNO = O.ORDERNO)
438 GROUP BY ORDERDATE;
439
440
441 /*12. 다음과 같은 신규 상품 정보를 products 테이블에 입력하시오.
442 "제품번호는 1007, 상품명은 목캔디, 단가는 3000원, 재고는 500개이다." */
443
444
445 SELECT *
446 FROM PRODUCTS;
447
448 INSERT INTO PRODUCTS (PNO, PNAME, COST, STOCK)
449 VALUES ('1007', '목캔디', '3000', '500');
450
451
452 /*13. 다음과 같은 4번 주문 정보를 입력하고, 재고를 적절히 수정하시오. 주문 정보를
453 작성한 문장으로 검색하면 다음과 같다.
454 "최진국(103)이 오늘 목캔디(1007)를 개당 3000원에 200개 주문하였으며, 배송지의
455 연락처는 352-4657이고, 결제가 완료된 상태이다." */
456
457 SELECT *
458 FROM ORDERS;
459
460 SELECT *
461 FROM ORDERDETAIL;
462
463 SELECT *
464 FROM CUSTOMERS;
465
466 SELECT *
467 FROM PRODUCTS;
468
469 INSERT INTO ORDERS (ORDERNO, ORDERDATE, ADDRESS, PHONE,
470 VALUES (4, SYSDATE, '제주 동광양', '352-4657', '결제완료', '103' );
471
472
473 INSERT INTO ORDERDETAIL (ORDERNO, PNO, QTY, COST)
474 VALUES (4, 1007, 200, 3000);
475
476 SELECT O.ORDERDATE, C.CNAME, C.ADDRESS, C.PHONE, O.STATUS,
477 FROM CUSTOMERS C
478 JOIN ORDERS O ON (C.CNO = O.CNO)
479 JOIN ORDERDETAIL D USING (ORDERNO)
480 JOIN PRODUCTS P USING (PNO);
481
482
483

```

484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523

000514 08 0001 7001 04 4 00